MyBatis注解处理流程:

在配置文件进行解析的时候进行注解逻辑处理。

|  |
| --- |
| SqlSessionFactoryBuilder  XMLConfigBuilder  XMLMapperBuilder  MapperRegistry  MapperAnnotationBuilder |

配置文件加载处理如下注解:

|  |
| --- |
| static {  SQL\_ANNOTATION\_TYPES.add(Select.class);  SQL\_ANNOTATION\_TYPES.add(Insert.class);  SQL\_ANNOTATION\_TYPES.add(Update.class);  SQL\_ANNOTATION\_TYPES.add(Delete.class);  SQL\_PROVIDER\_ANNOTATION\_TYPES.add(SelectProvider.class);  SQL\_PROVIDER\_ANNOTATION\_TYPES.add(InsertProvider.class);  SQL\_PROVIDER\_ANNOTATION\_TYPES.add(UpdateProvider.class);  SQL\_PROVIDER\_ANNOTATION\_TYPES.add(DeleteProvider.class);  } |

InputStream inputStream = Resources.getResourceAsStream("mybatis-config.xml");

SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);

XMLConfigBuilder parser = new XMLConfigBuilder(inputStream, environment, properties);

parser.parse()

parseConfiguration(parser.evalNode("/configuration"));

mapperElement(root.evalNode("mappers"));

|  |
| --- |
| <mappers>  <mapper resource="UserMapper.xml"/>  </mappers> |

XMLMapperBuilder mapperParser = new XMLMapperBuilder(inputStream, configuration, resource, configuration.getSqlFragments());

mapperParser.parse();

bindMapperForNamespace();

configuration.addMapper(boundType);

mapperRegistry.addMapper(type);

MapperAnnotationBuilder parser = new MapperAnnotationBuilder(config, type);

parser.parse();

parseStatement(method);

Method[] methods = type.getMethods();

for (Method method : methods) {

parseStatement(method);

}

SqlSource sqlSource = getSqlSourceFromAnnotations(method, parameterTypeClass, languageDriver);

Class<? extends Annotation> sqlAnnotationType = getSqlAnnotationType(method);

Class<? extends Annotation> sqlProviderAnnotationType = getSqlProviderAnnotationType(method);

Annotation sqlAnnotation = method.getAnnotation(sqlAnnotationType);

final String[] strings = (String[]) sqlAnnotation.getClass().getMethod("value").invoke(sqlAnnotation);

处理ResultMap注解:

MapperAnnotationBuilder类:

ResultMap resultMapAnnotation = method.getAnnotation(ResultMap.class);